Paralleled Fuzzy Time Series and Genetic Algorithm Prediction

Ankit Sirmorya, *CISE,* Preethu Thomas, *CISE,*
Shantanu Kande, *CISE,* and Sumeet Pande, *CISE*

Abstract—Paralleled Fuzzy Time Series(FTS) and Genetic algorithm(GA) prediction intends to bring these algorithms together to create a prediction model over time based events, while leveraging the MapReduce paradigm to enable parallel execution. Fuzzy Time Series is a algorithm that allows prediction by developing a model based on historical time based events. Genetic Algorithm emulates natural selection, such that we can improve the forecasts of a given prediction model. This process is run till the prediction converges sufficiently with the training data.

Index Terms—Fuzzy Time Series, Genetic algorithm, MapReduce

1 Introduction

ACCURATELY forecasting results for a given use case is always a challenge. It depends on multiple variables, there are exceptions which are not the norm that occur in everyday life and all related data for the given event may not be available.

An algorithm can make "as close to the truth" prediction, if it has sufficient historical data and it makes sense and relates the available data to the pattern of occurrences of that event. Weather forecasting, Election predictions, Stock market analysis and recommendations are few of the numerous examples that we see in everyday life, where prediction models are effectively used.

In this project, we intend to use Fuzzy Time Series (FTS) [1] and Genetic Algorithm (GA) [5] prediction models and translate them into MapReduce paradigm for reliable, accurate and efficient forecasting of time based events.

The novelty of this implementation is to provide an end-to-end solution by bringing together FTS, GA and MapReduce. FTS and GA being used in conjunction should improve prediction results, while MapReduce should improve running time as well as increase the amount of data ingested.

2 Related Work

Fuzzy Time Series (FTS) [1] is a prediction model that has proved its ability to forecast in numerous use cases, such as tourism [2], enrollments [3], temperature [4] and many more.

Holland's Genetic algorithm [5] has also been successfully been used in multiple prediction examples [6], [7], [11] and also in conjunction with FTS [8]. Genetic algorithm has also been run as MapReduce to solve the OneMax problem [10].

3 Architecture

For the purpose of implementation of our Hybrid Model we plan to predict the time-based events. As an input to our prediction model we have used the data pertaining to number of events per year for the previous years as our statistical data input.

The architecture is divided into three modules data ingestion, prediction and data visualization. This is as shown in *Figure 1*.

3.1 Data Ingestion

This phase will allow a configurable way of parsing the input data. Initially data will be read from a statistical data source such as publicly available data sets relevant to our prediction model. We have processed the data by

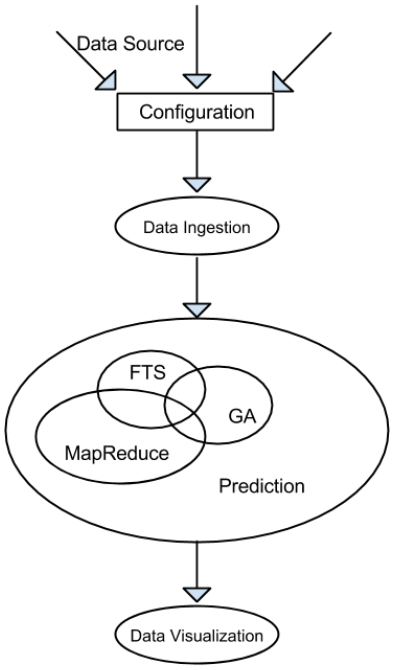

Fig. 1. Architecture of the implementation

application of MapReduce Paradigm so as to fine tune the data as per the input specification of GA+FTS Hybrid Prediction Model.

3.2 Prediction

The prediction module involves implementation of GA+FTS hybrid model on the incoming data from our data set. The subsystems for this model are explained below:

3.2.1 Genetic Algorithm (GA)

GA mimics the evolutionary principle of Survival of the Fittest. GA can be defined as a set of computational methods that are inspired by evolutionary biology such as mutation, selection, inheritance and crossover that can be used to find solutions to optimization problem. The first step of GA is to generate a random population of possible solution which correspond to a single generation. After this fitness value of every solution is calculated and multiple solutions are selected on basis of their fitness levels to generate a new generation. This process is carried on iteratively till a point of convergence is reached where there is no significant improvement in a generation over previous generation.

3.2.2 Fuzzy Time Series (FTS)

In our prediction model FTS is basically used as a fitness calculator that is to calculate correctness of prediction for each individual solution as a function of Root Mean Square Error (MSE).This is done by dividing the input data for time bound event into even and equal lengths of interval and then random selection of individuals that are product of GA iteration.

Thus the working of our hybrid GA+FTS prediction model can be visualized as combination of two functionalities where GA is used to improve correctness of each generation over previous generations and FTS is used as a fitness calculator to check the correctness of prediction in terms of mean square error.

3.3 Data Visualization

The result data visualization phase will use the prediction model results and display them in various output formats in order to easily understand and visualize the results.

4 Design Steps

4.1 Algorithm

The algorithm for this project:

Step 1: Read the training data which in this case will be time based events.

Step 2: Find the maximum and the minimum values of the training data and compute the Universe of Discourse.

Step 3: Create the first generation of Genetic Algorithm which will be a set of randomized individuals. Each of the individuals will be within the Universe of Discourse.

Step 4: Create the division of the generation data which can be used in the MapReduce model.

Step 5: The mapper task will apply Fuzzy Time Series Algorithm using the training data and GA population. This will make prediction for each individual and fitness value for each of them will be calculated.

Step 6: The reducer task will perform the selection, cross-over and mutation operations of GA based on the fitness value calculated by the mapper task. Thus, a new generation of individuals will be created.

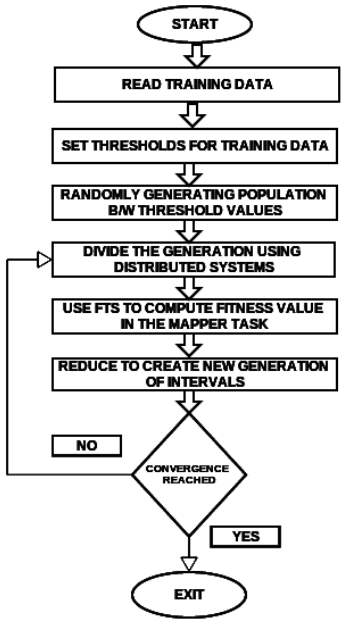

Fig. 2. Flowchart

Step 7: Repeat Steps 4 through 6 until the convergence point is not reached.

4.2 Flowchart

Figure 2 is the flowchart for this implementation

5 MODULES

There are three major modules namely data ingestion, prediction and data visualization.

5.1 Data ingestion

Figure 3 shows the interaction diagram of the Data Ingestion Module.

Data ingestion is used to consume raw data and produce output which will serve as the input to the prediction module. We have used MapReduce model to parse and raw data and generated the desired input. In this module we have collected time based data which is readily available. For example, the data format will be month wise data for the events for specific region starting from year 1971. The data parser module will use MapReduce paradigm to giveout as year wise format along with the minimum and maximum value from same.

The MapReduce framework operates on <key, value>pairs, that is, the framework accepts the input to the job as a set of <key, value>pairs and produces a set of <key, value>pairs as the output of the job, in different types. The <key, value >pair here will be <year, no. of events>. The function of Maps are to transform individual tasks that transform input records into intermediate records. Output of data maps, that is transformed intermediate records do not need to be of the same type as the input records. Reducer reduces a set of intermediate values which share a key to a smaller set of values. Sorted output of the mappers will be the input to reducer. The output of the reducer will be the sum of all events in a particular year.

5.2 Prediction

Figure 4, Figure 5, Figure 6, and Figure 7 are the class and interaction diagrams of the Prediction module.

Prediction module will include implementation of FTS and GA algorithms in the MapReduce paradigm. With the output from the data ingestion module, the minimum and maximum value as 440 and 1067. The universe of discourse U = [Dmin D1, Dmax + D2], where Dmin and Dmax denote the minimum events and the maximum events at particular region shown in *Figure 8*, and D1 and D2 are two proper values.

The universe of discourse U will be divided into several even and equal length intervals. Here we have selected D1 as 40 and D2 as 33. Then we will generate population between threshold values which will be our first generation.

FTS is used as a fitness calculator which calculates the correctness of prediction for each individual as a function of Mean Square Error (MSE). FTS fuzzifies the historical data of the time-based events and generates fuzzy logical relationships based on the order. GA will be used to improve correctness for each generation over the previous generation by making use of GA operators which use the calculated fitness values. The calculated MSE value of each derived chromosome obtained in the new generation, and the same cycle is

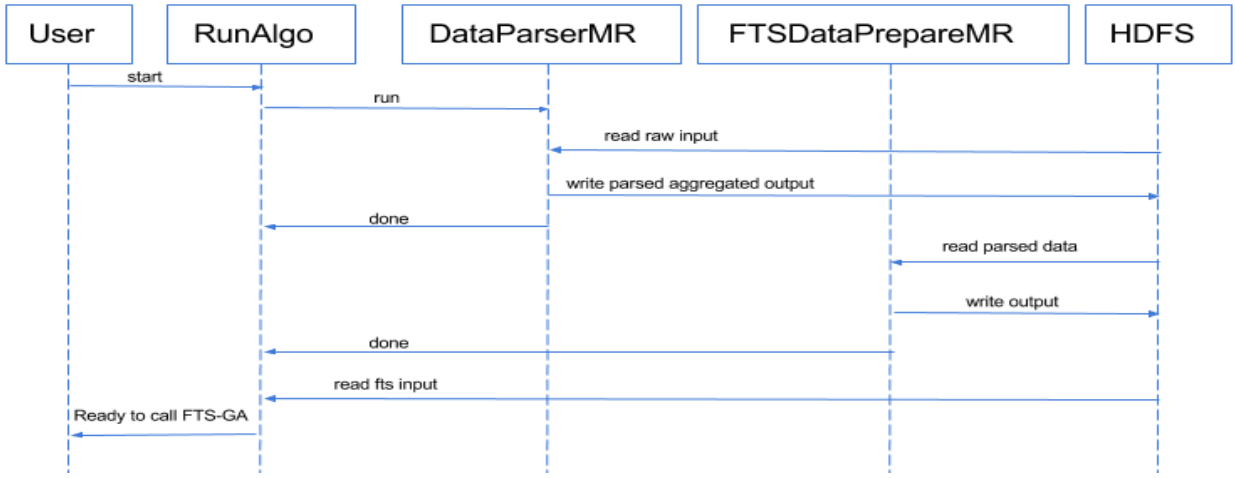

Fig. 3. Data Ingestion Interaction Diagram

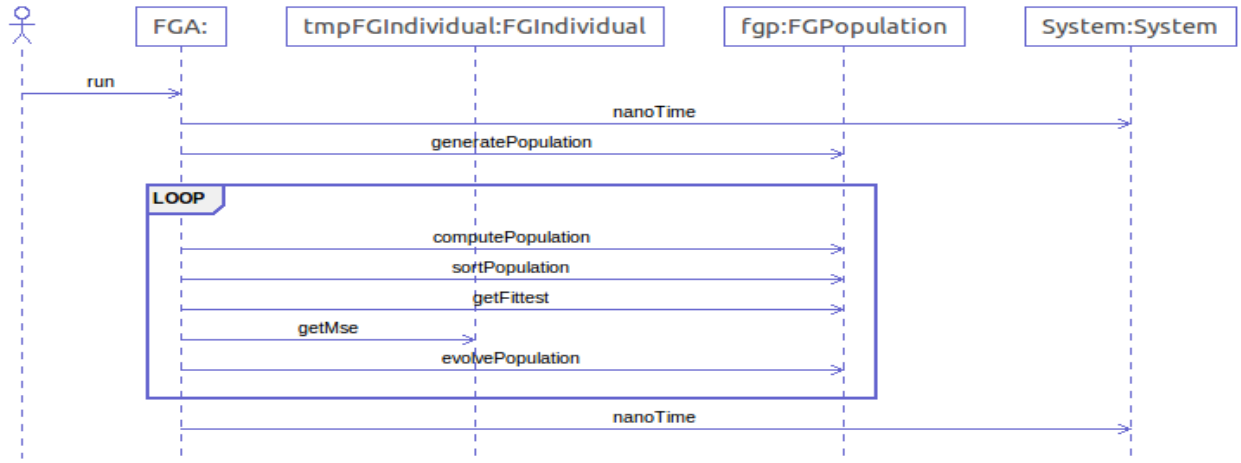

Fig. 7. Prediction module Interaction diagram

again implemented till a point of convergence is reached. In this case, the fittest individual will be the one having the lowest mean square error. If the minimum mean square error doesnt change over subsequent generations then the convergence point of the Genetic Algorithm is reached. *Fig. 5.* is the output of mean squared error between predicted and actual values.

5.3 Data Visualization

There are many different types of tools both web based and desktop based which will be very helpful for data visualization. We are exploring the use of Chart.js, Play framework and https://plot.ly/ (web based tool) to showcase our results.

Chart.js uses the HTML5 canvas element. Chart.js is supported in all modern web browsers, it is javascript based and is lightweight. Play framework is web application framework built using Scala. It is asynchronous and lightweight.

Plotly is web based tool in which we can upload data to create and style charts with Plotly's online spreadsheet. Using Plotly *Figure 10* is generated, which is the reduction of MSE over generations.

6 Results

The two main metrics for testing our forecasting models will be accuracy and running time. Higher accuracy would be indicated by comparing forecasted results with historical data, that is a lower Mean Squared Error between

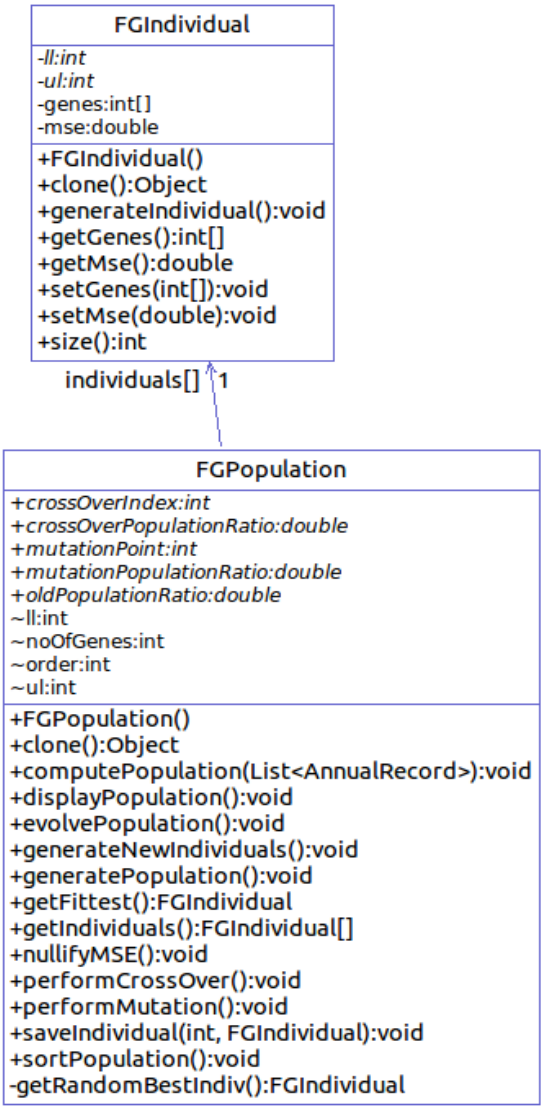

Fig. 4. Prediction module Class diagram - 1

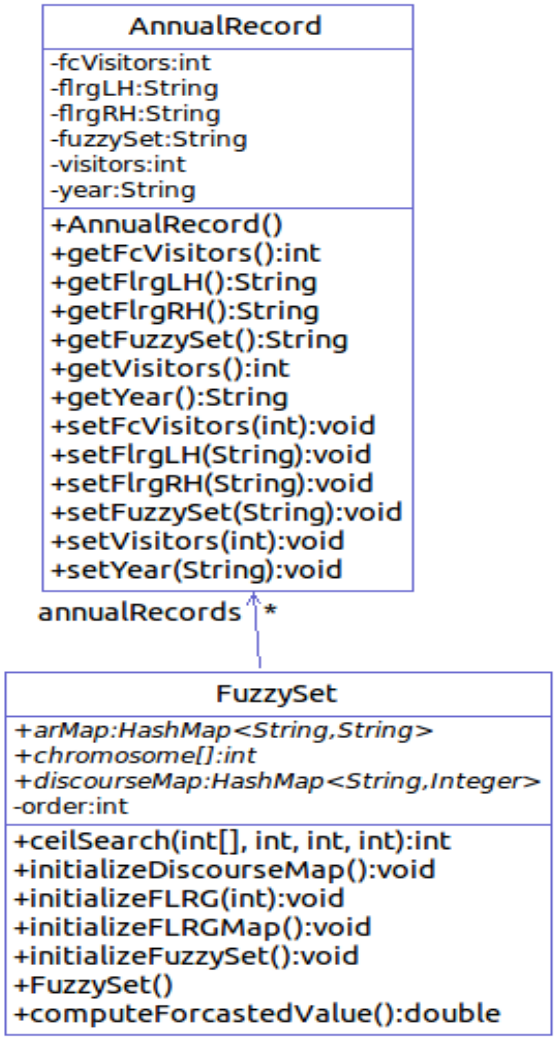

Fig. 5. Prediction module Class diagram - 2

predicted and actual data sets.

Another result that we intend to achieve is significant drop in running time by implementing the same in distributed/parallel (Hadoop Framework) manner over normal sequential manner.

Fig. 6. shows the expected differences in running time when the GA+FTS module is used in sequential manner and the same model being used by making use of MapReduce Paradigm. The results were generated on an Intel Core i3 @1.90GHz X4, 64 bits, 4GB RAM. We have observed some trends which are as follows, as number of intervals in the universe of discourse are increased, the MSE tends to decrease

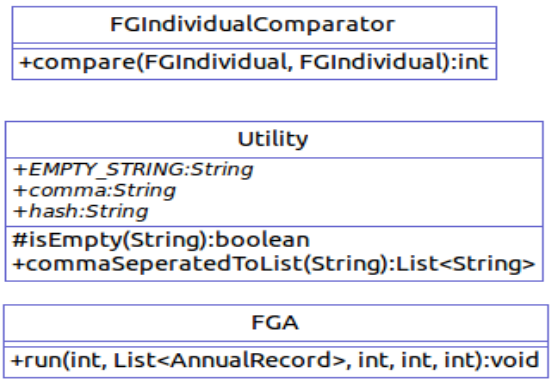

Fig. 6. Prediction module Class diagram - 3

```
1971    1025
1972    512
1973    1005
1974    852
1975    440
1976    502
1977    775
1978    465
1979    795
1980    970
1981    742
1982    635
1983    994
1984    759
1985    883
1986    599
1987    499
1988    590
1989    911
1990    862
1991    801
1992    1067
1993    917
Min 400
Max 1100
```

Fig. 8. Output of Data Ingestion

```
Generation : 85 = > 338.6084406314806
Generation : 86 = > 338.60841965705004
Generation : 87 = > 338.6083991648058
Generation : 88 = > 338.6083791383092
Generation : 89 = > 338.6083595618604
Generation : 90 = > 338.6083404204574
Generation : 91 = > 338.60832169975754
Generation : 92 = > 338.60830338604177
Generation : 93 = > 338.60828546618126
Generation : 94 = > 338.6082679276057
Generation : 95 = > 338.60825075827427
Generation : 96 = > 338.6082339466477
Generation : 97 = > 338.60821748166256
Generation : 98 = > 338.60820135270717
Generation : 99 = > 338.60818554959855
Generation : 100 = > 338.6081700625609
```

Fig. 9. Output of Mean Square Error over generations

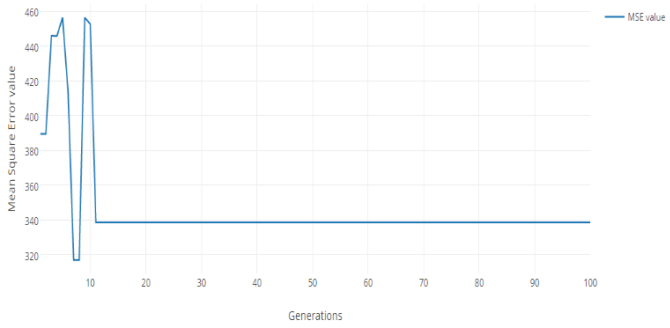

Fig. 10. Reducing Mean Squared Error over the generation

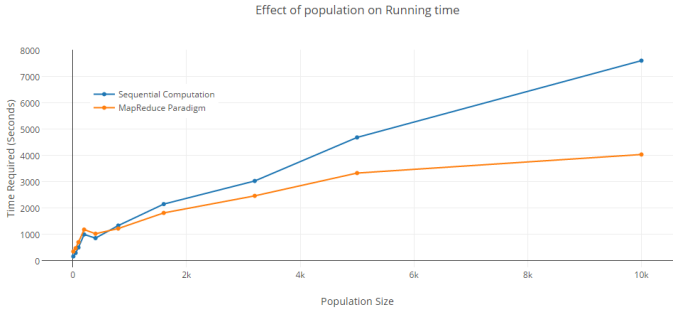

Fig. 11. Mock up of Speedup using MapReduce

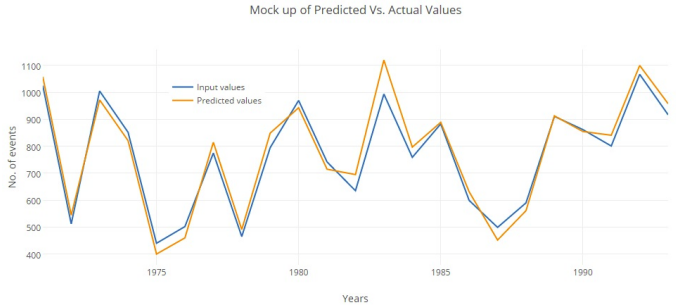

Fig. 12. Actual versus Mock Predicted values

and as the order of fuzzy logical relationships increases, the MSE will decrease.

7 Novelty of the Idea

Though there have been previous researches done in the field of parallel evolutionary algorithm [12], in this paper we have tried to overcome the time incurred by GA-FTS prediction model.

Also, with the advent of MapReduce [9] paradigm, it is much easier to abstract parallel system operations; while paving the way to concentrate on the problem at hand. However, a restrictive programming model like MapReduce, requires the translation of the algorithm in the form of map/reduce functions. This is still a major challenge and needs to be done with much consideration.

8 Possibilities to Consider

During the execution of the project, the following will still be in consideration:

Can we dynamically retrieve data? Initially, we will be manually downloading/creating the

data.

Can we use some other parallel framework? Possible candidates could be Spark [13], Storm [14], Giraph [15] or Hama [16]

Can we employ different MapReduce models of GA? Initially, only selection, crossover and mutation stages will be in reducer phase. We need to evaluate different mechanisms to improve the reduce phase.

Can we dynamically set thresholds for FTS/GA and number of individuals in GA? We will be manually tweaking these threshold values and visualizing the effects. We will try to have a dynamic way to sample training data and running multiple instance of the algorithm in parallel to decide appropriate threshold values.

REFERENCES

[1] Q. Song, B. Chissom, *Fuzzy time series and its models*, Fuzzy Sets and Systems, Vol 54, 1993, p. 269-277.

[2] C.H, Wang and L.C. Hsu, *Constructing and applying an improved fuzzy time series model: Taking the tourism industry for example*, Expert Systems with Applications, Vol 34, 2008, p. 2732-2738.

[3] S.M. Chen, *Forecasting enrollments based on high-order fuzzy time series*, Cybernetics and Systems: An International Journal, Vol 33, 2002, p. 1-16. 781-789.

[4] S.M. Chen and J.R. Hwang, *Temperature prediction using fuzzy time series*, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, Vol 30, Issue 2, 2000, p. 263-275.

[5] John H. Holland, *Adaptation in Natural and Artificial Systems*, 1975, University of Michigan Press, Ann Arbor.

[6] K.S. Shin and Y.J. Lee, *A genetic algorithm application in bankruptcy prediction modeling*, Expert Systems with Applications, 2002, 321-328

[7] A. Azadeh, S.F. Ghaderi, S. Tarverdian and M. Saberi. *Integration of artificial neural networks and genetic algorithm to predict electrical energy consumption*, Applied Mathematics and Computation, 2007, 186:1731-41

[8] S.M. Chen and N.Y. Chung, *Forecasting enrollments using high-order fuzzy time series and genetic algorithms*, Internat. J. Intell. Syst. 21 ,2006, 485-501.

[9] Jeffrey Dean and Sanjay Ghemawat, *MapReduce: Simplified Data Processing on Large Clusters*,OSDI'04: Sixth Symposium on Operating System Design and Implementation, 107-113.

[10] A. Verma, X. Llora, D.E. Goldberg, and R.H.Campbell, *Scaling genetic algorithms using mapreduce*, Intelligent Systems Design and Applications, 2009, 13-18.

[11] Deepak Singh and Ankit Sirmorya, *Article: Solving Real Optimization Problem Using Genetic Algorithm with Employed Bee (GAEB)*, International Journal of Computer Applications, 2012, 1-5.

[12] E. Cant-Paz, *A survey of parallel genetic algorithms*, Calculateurs paralleles, reseaux et systems repartis, 1998, 141-171.

[13] Apache Spark - *https://spark.apache.org/*

[14] Apache Storm - *https://storm.incubator.apache.org/*

[15] Apache Giraph - *http://giraph.apache.org/*

[16] Apache Hama - *https://hama.apache.org/*